PCT/NL 00 / 00720

NL00/720

KONINKRIJK DER NEDERLANDEN

Bureau voor de Industriële Eigendom

4

Hierbij wordt verklaard, dat in Nederland op 8 oktober 1999 onder nummer 1013249,

ten name van:

TRYLLIAN B.V.

te Amsterdam

een aanvrage om octrooi werd ingediend voor:

"Method for transferring a software module from a sender to a receiver in a computer system or network",

en dat de hieraan gehechte stukken overeenstemmen met de oorspronkelijk ingediende stukken.

Rijswijk, 15 november 2000

De Directeur van het Bureau voor de Industriële Eigendom,
voor deze,

b.a.

drs. N.A. Oudhof

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

1●13249

ABSTRACT

In a method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es), each class or group of classes is provided with a class identifier. Both the sender and receiver comprises a database of classes and groups of classes with corresponding class identifiers. The sender transmits the class identifier of a software module to be transferred to the receiver and the receiver checks its database for presence of the received class identifier. The receiver transmits a message "present" or "absent" to the sender and the sender transfers only the object of the software module or both the object and the class or group of classes depending on the presence or absence of the class or group of classes at the receiver.

7.II

1013249

NL2521-dV/jdh

Method for transferring a software module from a sender to a receiver in a computer system or network

The invention relates to a method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es).

In object-oriented software technology it is known to build a software module as a combination of so-called objects and classes, wherein the or each object that belongs to a class, is called an instance of the class. The objects generally contain only particular values for the variables specific to a predetermined software module, wherein the variables and methods to be carried out by the software module are defined in the class or classes.

With increasing use of computer systems and networks, such as the Internet, there is an increasing transfer of software modules of the object-oriented type between processes executed within one computer system of between computers of a computer network. This increasing transferring of software modules results in an increase of data traffic within the computer system of computer network.

A first object of the present invention is to provide a method of the above-mentioned type, wherein the data traffic within a computer system or computer network during transferring a software module is reduced.

It is a further object of the invention to provide a method of this type, wherein a software module can be transferred in a secure manner.

According to the invention a method of the above-mentioned type is provided, wherein each class or group of classes is provided with a class identifier, wherein both the sender and receiver comprises a database of classes and groups of classes with corresponding class identifiers, wherein the

8 U

sender transmits the class identifier of a software module to be transferred to the receiver and the receiver checks its database for presence of the received class identifier, wherein the receiver transmits a message "present" or "absent" to the sender and wherein the sender transfers only the object of the software module or both the object and the class or group of classes depending on the presence or absence of the class or group of classes at the receiver.

In this manner a method is obtained, wherein data traffic during transferring software modules is significantly reduced as the classes or groups of classes need not to be transferred in all transfers of software modules.

According to a favourable embodiment of the invention, the sender provides a class identifier by combining a given name of each class or group of classes of a software module and the result of a cryptographic hash function, wherein said result is obtained by executing a cryptographic hash function on the data file of the class or group of classes. In this manner a secure identifier is obtained, wherein errors due to identical identifiers for different classes or groups of classes are excluded.

According to a preferred embodiment, the receiver checks a class or group of classes received from a sender by comparing the result of the hash function of the received class identifier with the result obtained by carrying out the same cryptographic hash function on the data file of the class or group of classes received.

In this manner security in transferring software modules is guaranteed as receivers will refuse to use classes where the hash function result of the identifier does not match with the hash function result obtained by the receiver from the data file of the class or the group of classes received.

The invention will be further explained by reference to the drawings in which an embodiment of the method of the invention is schematically shown.

Fig. 1 shows in a schematic way a computer network in which a method of the invention is implemented.

Figs. 2 and 3 show flow diagrams of the operation of a sender and a receiver in the method of the invention.

A preferred embodiment of the method of the invention will be described as implemented by way of example in so-called software agents which can be used in a computer network, such as the Internet, for example for searching information on the Internet. However, it will be understood that the method of the invention is not restricted to this specific application. The method can be applied for transferring any software module of an object-oriented type as described.

Fig. 1 shows in a very schematic way the Internet comprising a number of interconnected computers or servers 1 and computers 2 of users. A user computer 2 can be connected to the Internet through a server 1 of an Internet service provider. If a user wishes to obtain information on a specific subject from the Internet, he can send a software agent with his request on the Internet to obtain this information. Finding adequate information on the Internet is a problem in view of the huge amount of information available on the Internet. Existing technology such as search engines store the most relevant information from the complete Internet to find this information in a quick manner if a user requests such information. Existing search engines show the disadvantage that searching the complete Internet from one location is time consuming so that such a search will not be carried out frequently and thereby the information stored by the search engine is dated fast. Moreover, existing search engines use their own criteria which do not necessarily correspond with the criteria which a user would use. A software agent searching the Internet with a request of its user may however use its own criteria for finding relevant information and the information can be obtained by communicating with other software agents also searching for information. During searching the Internet the software agents are transferred from one computer 1 to another computer 1 which causes a load of the network due to the data traffic involved in such transfers.

In order to reduce the load of the network caused by transferring the software agent, the following method is used.

It is noted that in the present description the term sender is used to indicate a computer 1 or 2 from which a software agent is to be transferred to another computer 1 or 2 which receiving computer is called receiver. It is noted however that the present method can also be used to transfer an agent being present in a directly executable format in a process executed in one computer to another process executed in the same computer. In this case the terms sender and receiver refer to such processes between which a software agent is transferred. It is further noted that in an application of the method in a computer network, such as the Internet, it is not necessary that all computers of the network are adapted or programmed to operate as sender and/or receiver.

Each software agent is made as a software module of an object-oriented type. This means within the present description that the software module comprises one or more objects and one or more classes, wherein the objects are instances of the class or classes. An object comprises data, such as variable values. A class comprises definitions of the data structure, i.e. name and type of data fields, and definitions of functions or routines which are carried out on the objects of the class. Objects or more particular the variable values of the objects often change due to the execution of the routines which are defined in the class. If a software module comprises two or more classes these classes will be indicated as a group of classes in this specification.

According to the method described, a software module is transferred from a first computer 1, the sender, to a second computer 1, the receiver, in the computer network, by transferring the class or classes of the software module only if the class or classes are not present at the receiver side. In those computers of the network adapted to function as a sender and/or receiver a database is provided for storing classes or groups of classes, wherein a unique class identifier is added to each class or group of classes. When a sender wishes to transfer a software module to a receiver, the sender and receiver start to communicate to check or whether or not the classes or group of classes of the software module to be transferred is present in the receiver. The steps of the method are schematically shown in figs. 2 and 3 for the sender and receiver sides, respectively.

To start the transfer protocol, the sender transmits a start message to the receiver indicating "hereinafter follow all data of a software module". All objects which are part of the software module to be transferred are combined in a data file and this data file is transmitted to the receiver. Further, the sender transmits the unique class identifier to the receiver. Thereafter, the receiver checks its database for the presence of the unique class identifier received. The receiver transmits a return message "present" or "absent" to the sender and the sender transmits a data file with the class or group of classes only if a message "absent" was received. If Java is used as software language for the software module, the data files of the classes of one software module can be bundled into one data file by means of a Java tool JAR. The thus obtained data file is a so-called jar-file. Generally in the present method a class will contain a software routine or the like. Other embodiments could involve using classes containing reference to routines or the like and not the routine itself.

The receiver obtains the software module by combining the objects received from the sender with the class or group of classes either retrieved from its database or received from the sender. If the receiver succeeds in combining the objects and class or group of classes, the receiver transmits a message "transfer succeeded" to the sender. If the receiver cannot combine the objects and classes, a message "transfer not succeeded" will be transmitted to the sender.

In this manner data traffic in the network will be significantly reduced as generally it will only be necessary to transfer the objects of a software module.

Of course, at any transfer of a class or group of classes, a receiver receiving new classes can store the class or group of classes together with the corresponding unique class identifier in its database for future use. In this manner the database will gradually be further completed reducing future data traffic.

It is noted that although in the above-described example, the objects and class identifier are transmitted from the sender to the receiver without interruption, its is also possible to transmit first the class identifier to check its presence at the receiver and thereafter the objects. However, the order described shows the advantage that the number of switchings from transmitting to receiving at the sender and receiver sides is reduced resulting in a further reduction of the time period for transferring a software module as switching in a computer network is relatively time consuming.

In the method of the invention the unique class identifier is preferably obtained in the following manner. First the user may give a predetermined name to a class or group of classes. This given name is a first part of the class identifier. Further, all classes of a software module are bundled into one data file which is the data file to be transferred if the group of classes is absent at the receiver side. At the sender side a program is executed determining a cryptographic hash function of the data file of the group of classes and the result of this hash function is stored. This hash function result is the second part of the unique class identifier.

It is noted that any cryptographic hash function can be used which provides a result which is significantly shorter than the original data file, wherein a small change in the original data file provides a large change in the hash function result and wherein it is very difficult to determine an input data file leading to a predetermined hash function result.

As a further option to provide a unique class identifier, it is possible to add the length of the data file as a third part to the unique class identifier.

In this manner it is guaranteed that if a sender transmits the unique class identifier of a software module to a receiver and the receiver indicates that the class or group of classes of the unique class identifier received is present in its database, indeed the correct class or classes are available at the receiver side.

Further, the use of a cryptographic hash function provides security to the transfer method. The receiver will always check the unique class identifier with the data file of classes received. To this end the receiver will execute the same cryptographic hash function on the data file of the classes and will compare the result of its own hash function with the hash function result in the class identifier. If a match is not found, the receiver will transmit a message "transfer not succeeded" to the sender and will terminate the transfer protocol.

The method described can be used advantageously in an Internet application of software agents, wherein the software agent travels on the Internet to search information and the like. However, it will be understood that the method results in the same advantages of reducing data traffic between processes running in one computer system or the transfer of software modules between computers in any type of network.

The invention is not restricted to the above described embodiment and can be varied in a number of ways within the scope of the following claims.

CLAIMS

1. Method for transferring a software module from a sender to a receiver in a computer system or network, wherein the software module comprises at least one object and at least one class, the object being an instance of the class(es), wherein each class or group of classes is provided with a class identifier, wherein both the sender and receiver comprises a database of classes and groups of classes with corresponding class identifiers, wherein the sender transmits the class identifier of a software module to be transferred to the receiver and the receiver checks its database for presence of the received class identifier, wherein the receiver transmits a message "present" or "absent" to the sender and wherein the sender transfers only the object of the software module or both the object and the class or group of classes depending on the presence or absence of the class or group of classes at the receiver.

2. Method according to claim 1, wherein the sender transmits first all objects and the class identifier to the receiver, wherein the sender transmits the class or group of classes to the receiver if a message "absent" is received.

3. Method according to claim 1 or 2, wherein the receiver obtains the software module to be transferred by combining the object received with the class or group of classes retrieved from its database or received, wherein the receiver transmits a message "transfer succeeded" or "transfer not succeeded" depending on whether or not the receiver succeeds in combining the object and class or group of classes.

4. Method according to any one of the preceding claims, wherein the receiver stores each class and group of classes with the corresponding class identifier received in its database for later use.

5. Method according to any one of the preceding claims, wherein the sender provides a class identifier by combining a given name of each class or group of classes of a software module and the result of a cryptographic hash function, wherein said result is obtained by executing a crypto-

graphic hash function on the data file of the class or group of classes.

6. Method according to claim 5, wherein a sender further combines the length of the data file of the class or group of classes with the given name and the result of the hash function to provide the class identifier.

7. Method according to claim 5 or 6, wherein the receiver checks a class or group of classes received from a sender by comparing the result of the hash function of the received class identifier with the result obtained by carrying out the same cryptographic hash function on the data file of the class or group of classes received.

8. Method according to claim 7, wherein the receiver transmits a message "transfer succeeded" or "transfer not succeeded" depending on the comparison of the result of the hash function on the data file received and the result of the hash function of the class identifier.

9. Method according to any one of the preceding claims, wherein senders and receivers are computers in computer network, such as the Internet.

10. Method according to claim 9, wherein the software module is a so-called agent for searching, exchanging and/or providing information on the network.

1013249

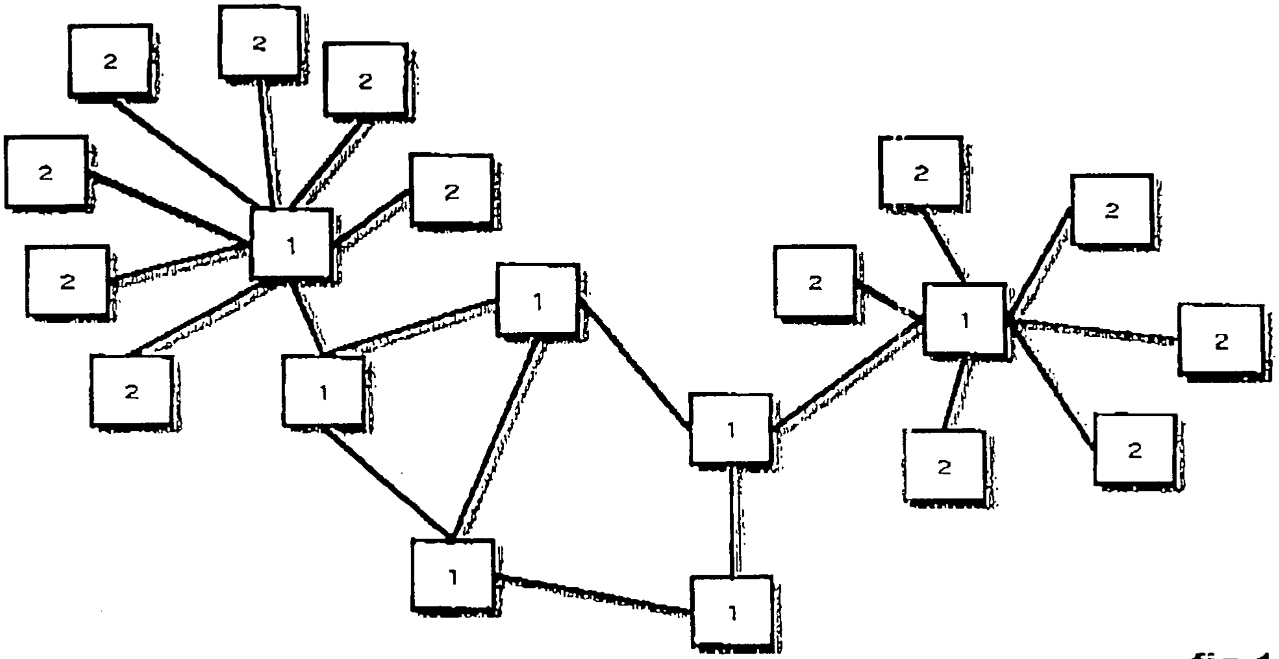

fig.1

1013249

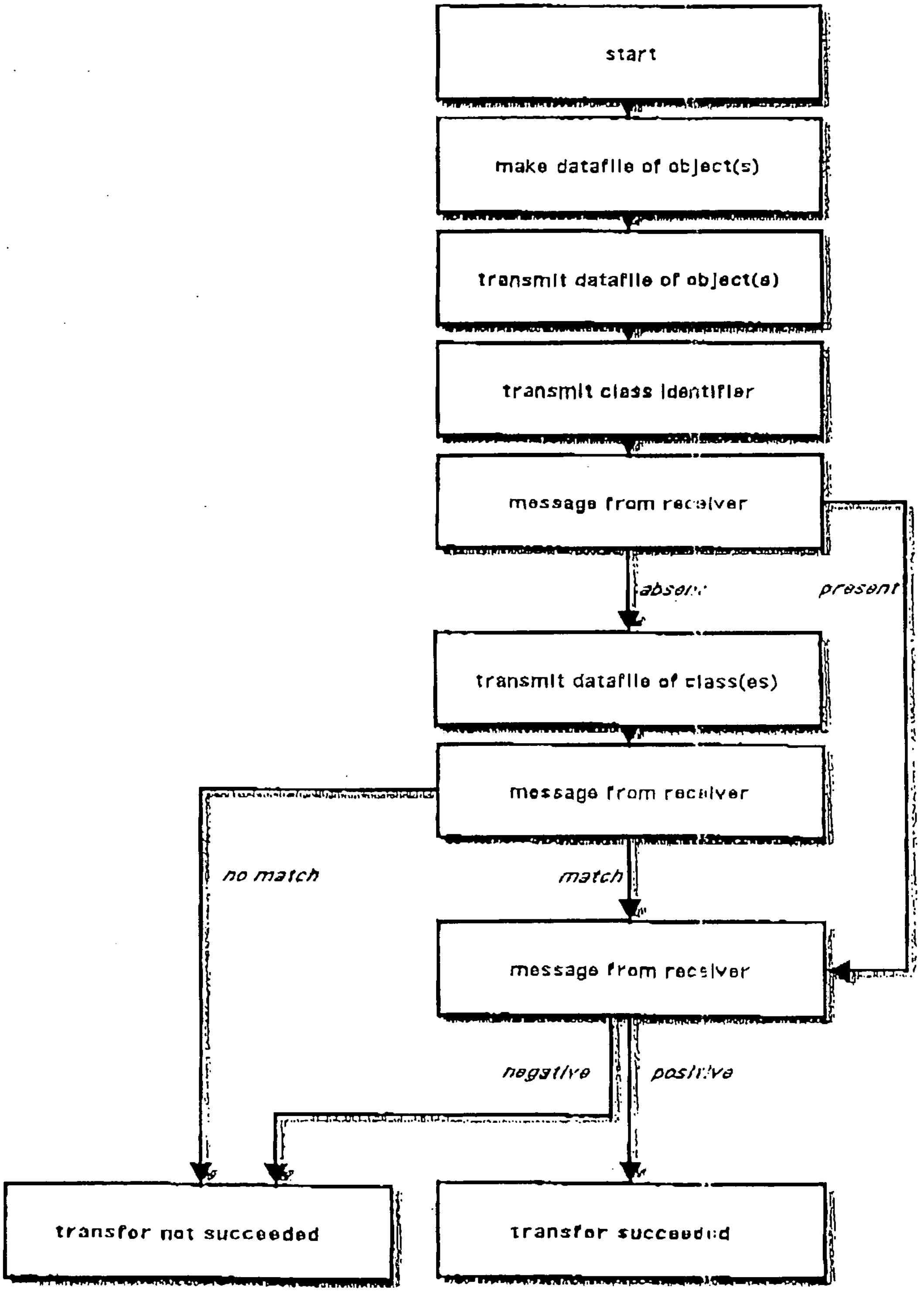

fig.2

9 III b

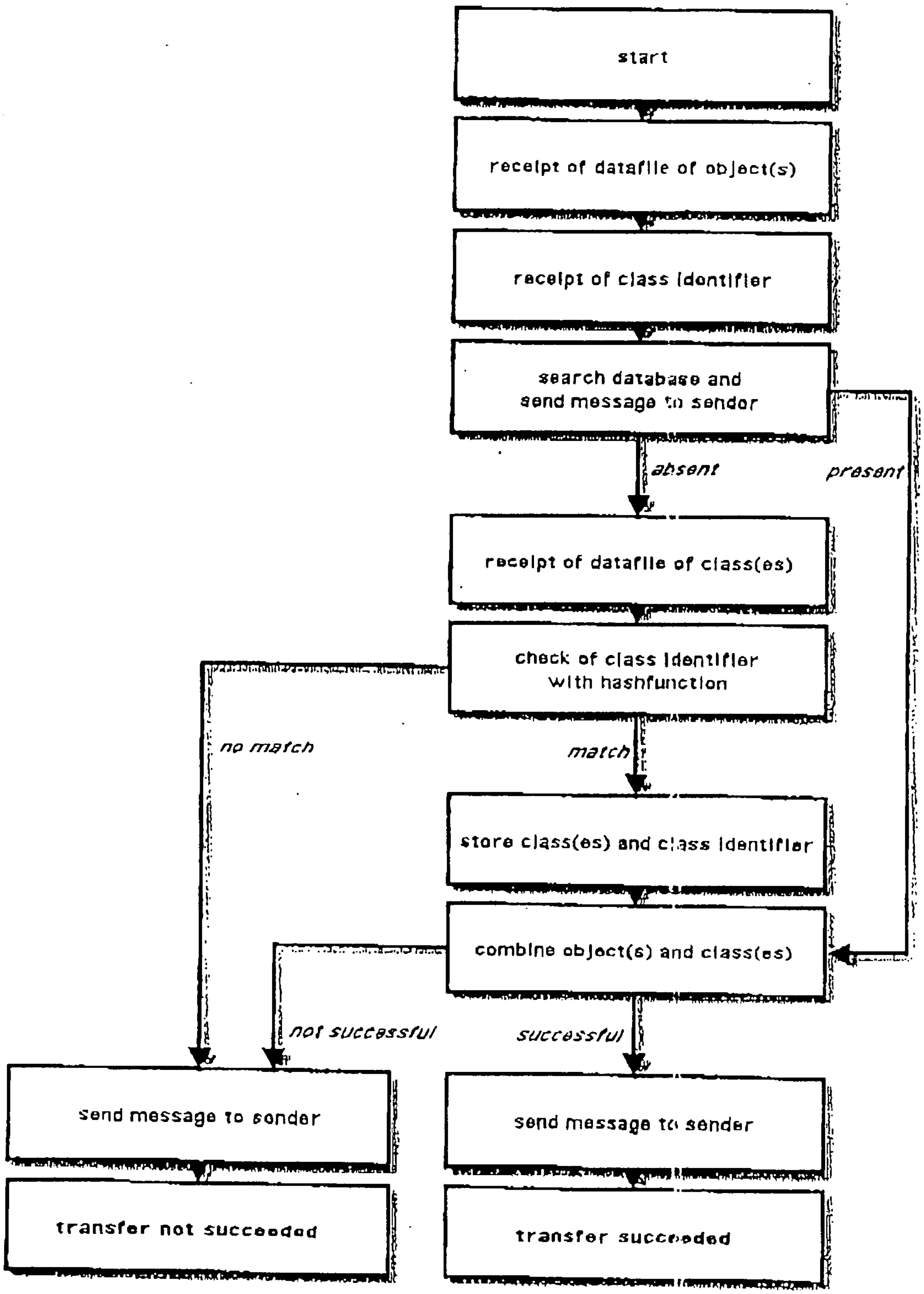
start
receipt of datafile of object(s)
receipt of class identifier
search database and send message to sendor
absent
present
receipt of datafile of class(es)
check of class identifier with hashfunction
no match
match
store class(es) and class identifier
combine object(s) and class(es)
not successful
successful
send message to sender
transfer not succeeded
send message to sender
transfer succeeded

fig.3

9 III c